AMENDMENTS TO THE CLAIMS

The listing of claims below replaces all prior versions, and listings, of claims:

1. (Previously Presented) A test system comprising:

an emulation module to receive environment information of a database system separate from the test system, the emulation module to emulate an environment of the database system based on the environment information;

a first module executable in the emulated environment and adapted to receive a set of queries and to provide a set of candidate indexes for the set of queries, the first module adapted to eliminate one or more candidate indexes based on one or more predetermined criteria; and

a second module executable in the emulated environment and adapted to generate a recommended index from the set of candidate indexes.

2. (Previously Presented) The test system of claim 1, wherein the set of queries comprises a set of SQL statements.

3. (Previously Presented) The test system of claim 1, wherein the second module is adapted to generate at least another recommended index from the set of candidate indexes.

4. (Previously Presented) The test system of claim 1, wherein the second module comprises an optimizer that is adapted to use statistics.

5. (Previously Presented) The test system of claim 4, wherein the statistics are based on a scan of a sample of one or more tables, the sample less than all the rows of the one or more tables.

6. (Previously Presented) The test system of claim 5, further comprising a graphical user interface to receive an indication of a user-specified size of the sample.

7. (Previously Presented) A system comprising:

a first module adapted to receive a set of queries and to provide a set of candidate indexes for the set of queries, the first module adapted to eliminate one or more candidate indexes based on one or more predetermined criteria; and

an optimizer adapted to generate a recommended index from the set of candidate indexes,

wherein the one or more predetermined criteria comprises a threshold change rate, the first module adapted to eliminate one or more candidate indexes having a change rate exceeding the threshold change rate.

8. (Original) The system of claim 7, wherein the first module is adapted to further eliminate a candidate index that is a subset of another candidate index.

9. (Previously Presented) The test system of claim 1, wherein the second module comprises an analysis module and an optimizer, the analysis module adapted to apply a genetic algorithm, the analysis module adapted to cooperate with the optimizer to generate the recommended index using the genetic algorithm.

10. (Previously Presented) The test system of claim 9, wherein the first module is adapted to provide the set of candidate indexes by identifying the candidate indexes from the set of queries and defining the set of queries in a database.

11. (Previously Presented) The test system of claim 10, wherein the analysis module is adapted to access the database to retrieve the candidate indexes.

12. (Previously Presented) The test system of claim 10, further comprising a validation module adapted to validate the recommended index in a database system.

13. (Previously Presented) The test system of claim 12, further comprising a user interface to receive user-specified one or more indexes, the optimizer adapted to generate a cost associated with a query plan based on the user-specified one or more indexes.

14. (Previously Presented) The test system of claim 13, wherein the user interface is adapted to receive a user-specified percentage value, the system further comprising another module to collect statistics based on a sample of rows of one or more tables, a size of the sample based on the user-specified percentage value.

15. (Previously Presented) The test system of claim 14, further comprising another module adapted to provide a hint on which table or tables statistics need to be collected.

16. (Previously Presented) The test system of claim 10, wherein the analysis module is adapted to access the database to retrieve the candidate indexes.

17. (Previously Presented) The test system of claim 1, wherein the second module comprises an analysis module and an optimizer, the analysis module adapted to apply a predetermined algorithm, the analysis module adapted to cooperate with the optimizer to generate the recommended index using the predetermined algorithm.

18. (Previously Presented) The test system of claim 17, wherein the analysis module is adapted to submit candidate indexes to the optimizer, the optimizer adapted to determine the cost of one or more of the queries based on the candidate indexes.

19. (Previously Presented) The test system of claim 18, wherein the optimizer is adapted to select the candidate index associated with a lowest cost as the recommended index.

20. (Previously Presented) The test system of claim 1, wherein the set of queries comprises a workload captured from the database system, and wherein the database system is a parallel system having plural access modules, the environment information containing information regarding the parallel system and plural access modules.

21. (Previously Presented) The test system of claim 20, wherein the optimizer is adapted to compute costs for the candidate indexes in the emulated environment of the database system.

22. (Withdrawn) A method of selecting a recommended index for a database system, comprising:

receiving a workload containing a set of queries of the database system, the database system being a parallel database system having a plurality of access modules and storage modules, the access modules to manage parallel access of tables in corresponding storage modules;

generating a set of candidate indexes from the workload;

removing candidate indexes based on one or more predetermined criteria; and

invoking an optimizer to provide cost analysis for the set of candidate indexes in the parallel database system to generate the recommended index from the set of candidate indexes.

23. (Withdrawn) The method of claim 22, further comprising applying a predetermined algorithm to identify the recommended index.

24. (Withdrawn) The method of claim 23, wherein applying the predetermined algorithm comprises applying a genetic algorithm.

25. (Withdrawn) The method of claim 24, further comprising selecting the candidate index having a lowest cost as the recommended index.

26. (Withdrawn) The method of claim 22, further comprising providing graphical user interface screens to receive user input for selecting the recommended index.

27. (Withdrawn) A method of selecting a recommended index for a database system, comprising:

receiving a workload containing a set of queries of the database system;

generating a set of candidate indexes from the workload;

removing candidate indexes based on one or more predetermined criteria;

invoking an optimizer to provide cost analysis to generate the recommended index from the set of candidate indexes; and

providing graphical user interface screens to receive user input for selecting the recommended index,

wherein providing graphical user interface screens comprises displaying an activatable item to perform workload identification to identify the workload.

28. (Withdrawn) The method of claim 27, wherein providing the graphical user interface screens further comprises displaying another activatable item to perform workload definition to save the workload into a database.

29. (Withdrawn) The method of claim 28, wherein providing the graphical user interface screens further comprises displaying another activatable item to perform index analysis to analyze the candidate indexes to generate the recommended index.

30. (Withdrawn) The method of claim 29, wherein providing the graphical user interface screens further comprises displaying another activatable item to validate the recommended index in the database system.

31. (Withdrawn) The method of claim 29, wherein providing the graphical user interface screens comprises displaying another activatable item to validate the recommended index in a test system having an emulated environment of the database system.

32. (Withdrawn) The method of claim 30, wherein providing the graphical user interface screens further comprises displaying another activatable item to cause submission of a command to the database system to create the recommended index.

33. (Withdrawn) The method of claim 26, wherein providing the graphical user interface screens comprises displaying one or more reports relating to the recommended index.

34. (Withdrawn) The method of claim 33, wherein providing the graphical user interface screens further comprises displaying a comparison of a cost using the recommended index with a cost using an existing index.

35. (Withdrawn) The method of claim 33, wherein providing the graphical user interface screens further comprises displaying cost improvement relating to use of the recommended index.

36. (Withdrawn) The method of claim 22, wherein invoking the optimizer is performed in a test system separate from the database system.

37. (Withdrawn) A method of selecting a recommended index for a database system, comprising:

receiving a workload containing a set of queries of the database system;

generating a set of candidate indexes from the workload;

removing candidate indexes based on one or more predetermined criteria;

invoking an optimizer to provide cost analysis to generate the recommended index from the set of candidate indexes,

wherein invoking the optimizer is performed in a test system separate from the database system; and

importing environment information of the database system into the test system to emulate the database system in the test system.

38. (Withdrawn) The method of claim 37, wherein importing the environment information comprises importing the environment information of a parallel database system having plural access modules.

39. (Withdrawn) The method of claim 22, wherein invoking the optimizer is performed in the database system.

40. (Withdrawn) An article comprising at least one storage medium containing instructions that when executed cause a system to:

receive a set of queries;

generate a set of candidate indexes from the set of queries;

eliminate candidate indexes based on one or more predetermined criteria;

invoke an optimizer to perform cost analysis of the candidate indexes; and

use the cost analysis to select a recommended index for a database system,

wherein eliminating candidate indexes based on one or more predetermined criteria comprises at least one of:

eliminating candidate indexes that are changed with updates at a rate greater than a predetermined change rate threshold; and

eliminating a candidate index that is a subset of another candidate index.

41. – 42. (Cancelled)

43. (Withdrawn) The article of claim 40, wherein the instructions when executed cause the system to apply a genetic algorithm to select the recommended index.

44. (Withdrawn) The article of claim 40, wherein the system is a test system separate from the database system, the instructions when executed causing the test system to:

import environment information regarding the database system;

emulate an environment of the database system based on the imported environment information,

wherein the generating, eliminating, invoking, and using acts are performed in the emulated environment.

45. (Withdrawn) The article of claim 44, wherein the environment information comprises cost-related information, statistics, and random samples from the database system.

46. (Previously Presented) The article of claim 1, wherein the environment information comprises cost-related information, statistics, and random samples from the database system.